UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/052,039	01/17/2002	Shishir Pardikar	13768.783.91	1257

47973 7590 02/05/2008
WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
SERRAO, RANODHI N	
ART UNIT	**PAPER NUMBER**
2141	

MAIL DATE	DELIVERY MODE
02/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

AK

Office Action Summary	Application No.	Applicant(s)	
	10/052,039	PARDIKAR ET AL.	
	Examiner	Art Unit	
	Ranodhi Serrao	2141	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18,21-24 and 26-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18, 21-24, 26-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 05 December 2007 have been fully considered but they are not persuasive.

2. Applicant argued,

> *No mention is made of determining which redirectors are configured to handle a file request. Indeed, this does not need to be done in Oehrke, as it is repeatedly mentioned that the application processors provide the same services (Col. 2:65, Col. 8:40, Claim 1, etc.).*

3. Examiner points out that Serlet et al. is cited to teach the limitation of, determining which redirectors are configured to handle a file request. As explained in col. 6, lines 25-64 of Serlet, requests are sent to the WebDAV/HTTP servers and responses are returned which may include a status code indicating whether or not the server can handle the requests. See below rejection.

4. Applicant further stated,

> *Furthermore, no mention is made of precedence, priority order, or determining which redirector has precedence to handle an incoming request based on a stored priority order. Oehrke makes determinations as to which application processor should process the request based on weighted statistics gathered by the redirectors, not based on a stored priority order.*

5. In col. 8, lines 29-62, Oehrke states, "The statistical information collected by the parameter redirectors 52 is applied to a function and various weights. The result of the application to the function is then compared for each of the various local redirectors 56. The comparison is used to select the potentially most responsive local redirector 56." Selecting the potentially most responsive local redirector is obviously the same as

giving precedence to it. It is inherent that a priority order must be stored in order to utilize this selection system. And in col. 9, lines 52-64, Oehrke describes maintaining tables of addresses associated with redirectors that may be updated. Therefore a priority order is clearly stored, and Oehrke teaches the claimed invention.

6. Examiner directs applicant's attention to the claim language, maintaining at an I/0 manager a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a WebDAV I/0 request **in the event that** two or more suitably configured redirectors respond to a WebDAV I/0 request, each redirector indicating a configuration suitable for handling the I/0 request; Emphasis added. This is a conditional statement that does not need to be given any weight because if there is only one redirector that responds to the request, a priority order does not need to be maintained. There is no explanation of what steps need to occur otherwise. The claim language is broad enough to be interpreted in this manner by one of ordinary skill. Therefore this entire limitation has no effect on the claimed invention.

7. The examiner points out that the pending claims must be "given the broadest reasonable interpretation consistent with the specification" [In re Prater, 162 USPQ 541 (CCPA 1969)] and "consistent with the interpretation that those skilled in the art would reach" [In re Cortright, 49 USPQ2d 1464 (Fed. Cir. 1999)]. In conclusion, upon taking the broadest reasonable interpretation of the claims, the cited references teach all of the claimed limitations. And the rejections are maintained. See below.

Claim Objections

8. Claims 1, 16, and 33 are objected to because of the following informalities: Claim 1 recites, "maintaining at an I/O manager a stored a priority order" in line 3. This phrase is grammatically incorrect. Claims 16 and 33 recite similar language. Appropriate correction is required.

Claim Rejections - 35 USC § 103

9. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

10. Claims 1-11, 15-18, 23, 26-27, and 32-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Serlet et al. (6,842,770) and Oehrke et al. (6,735,631).

11. As per claim 1, Serlet et al. teaches in a computer network, a method of automatically and transparently handling WebDAV server and file access requests (see Serlet et al., col. 5, line 60-col. 6, line 14), the method comprising: two or more suitably configured redirectors capable of responding to a WebDAV I/O request (see Serlet et al., col. 6, lines 25-64: wherein the response from the WebDAV/HTTP servers reveals capability status); receiving at the I/O manager an WebDAV I/O request initiated from an application program, wherein the request indicates a path and filename of a remote file accessible via WebDAV (see Serlet et al., col. 6, lines 25-64); downloading the file to a local cache of the redirector's file system (see Serlet et al., col. 9, lines 54-63), and returning a file handle corresponding to the file in the local cache to the application program (see Serlet et al., col. 11, lines 24-49); providing access to the file in the local

cache of the file system via the file handle (see Serlet et al., col. 9, line 64-col. 10, line 13); and receiving a request to close the file via the file handle, and when received, uploading the file from the local cache of the file system to the WebDAV server (see Serlet et al., col. 12, lines 45-54). But fails to teach maintaining at an I/O manager a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a WebDAV I/O request, each redirector indicating a configuration suitable for handling the I/O request; polling available redirectors to determine which redirectors are configured to handle the application program's WebDAV I/O file request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting WebDAV file requests to corresponding WebDAV server computer systems that store the remote files; receiving responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received WebDAV I/O file request; determining from the stored priority order which of the plurality of suitably configured redirectors has precedence to handle the WebDAV I/O request; based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the WebDAV I/O request. However, Oehrke et al. teaches maintaining at an I/O manager a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O request, each redirector indicating a configuration suitable for handling the I/O request (see Oehrke et al., col. 8, lines 29-62); polling available redirectors to determine which redirectors are configured to handle the application program's I/O file request,

each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting file requests to corresponding server computer systems that store the remote files (see Oehrke et al., col. 8, lines 29-62); receiving responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received I/O file request (see Oehrke et al., col. 8, line 63-col. 9, line 2); determining from the stored priority order which of the plurality of suitably configured redirectors has precedence to handle the I/O request (see Oehrke et al., col. 8, lines 29-62); based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the I/O request (see Oehrke et al., col. 8, line 63-col. 9, line 2). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. to maintaining at an I/O manager a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O request, each redirector indicating a configuration suitable for handling the I/O request; polling available redirectors to determine which redirectors are configured to handle the application program's I/O file request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting file requests to corresponding server computer systems that store the remote files; receiving responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received I/O file request; determining from the stored priority order which of the plurality of suitably configured redirectors has

precedence to handle the I/O request; based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the I/O request in order to use redirectors to re-route traffic to other application processors when one processor is unavailable and load balance between available processors (see Oehrke et al., abstract).

12. As per claim 2, Serlet-Oehrke teach receiving an I/O request initiated from an application program comprises, receiving a Universal Resource Identifier corresponding to a file on the WebDAV server (see Serlet et al., column 9, lines 38-53).

13. As per claim 3, Serlet-Oehrke teach wherein receiving an I/O request initiated from an application program comprises, receiving a filename and an identifier previously mapped to a share on the WebDAV server (see Serlet et al., column 9, lines 54-63).

14. As per claims 4, 5, 7, 9, and 11, the above-mentioned motivation of claim 1 applies fully in order to combine Serlet et al. and Oehrke.

15. As per claim 4, Oehrke et al. teaches a method wherein polling available redirectors to determine which redirectors are configured to handle the application program's I/O request (see Oehrke et al., col. 8, lines 29-62) and Serlet et al. teaches, issuing an HTTP OPTIONS request, and evaluating a response therefrom (see Serlet et al., column 7, lines 35-56: wherein requests to create or delete a file or directory, etc. serve the function of HTTP OPTIONS request).

16. As per claim 5, Oehrke et al. teaches a method wherein polling available redirectors to determine which redirectors are configured to handle the application program's I/O request (see Oehrke et al., col. 8, lines 29-62) and Serlet et al. teaches

issuing a WebDAV PROPFIND request directed to share on the WebDAV server, and evaluating a response therefrom (see Serlet et al., column 11, lines 24-49).

17. As per claim 6, Serlet-Oehrke teach a method wherein the WebDAV server returns property information in response to the WebDAV PROPFIND request directed to the share and further comprising, maintaining the property information in a local data structure (see Serlet et al., column 8, line 66-column 9, line 22; column 11, lines 50-65).

18. As per claim 7, Oehrke et al. teaches a method wherein polling available redirectors to determine which redirectors are configured to handle the application program's I/O request (see Oehrke et al., col. 8, lines 29-62) and Serlet et al. teaches issuing a WebDAV PROPFIND request directed to on the WebDAV server, and evaluating a response therefrom (see Serlet et al., column 11, lines 24-49).

19. As per claim 8, Serlet-Oehrke teach a method wherein the WebDAV server returns property information in response to the WebDAV PROPFIND request directed to the file, and further comprising, maintaining the property information in a local data structure (see Serlet et al., column 8, line 66-column 9, line 22; column 11, lines 50-65).

20. As per claim 9, Oehrke et al. teaches a method wherein polling available redirectors to determine which redirectors are configured to handle the application program's I/O request (see Oehrke et al., col. 8, lines 29-62) and Serlet et al. teaches issuing an HTTP OPTIONS request, evaluating a corresponding response, and determining that the server a WebDAV server (see Serlet et al., column 6, line 25-64); issuing a WebDAV PROPFIND request directed to a share on the WebDAV server, evaluating a corresponding response, and determining that the share exists on the

WebDAV server, the response including share property information; and issuing a WebDAV PROPFIND request directed to the file, evaluating a corresponding response, and determining that the file exists, the response including file property information (see Serlet et al., column 11, lines 24-49).

21. As per claim 10, Serlet-Oehrke teach a method of maintaining the share property information and the file property information in at least one local data structure (see Serlet et al., column 8, line 66-column 9, line 22; column 11, lines 50-65).

22. As per claim 11, Oehrke et al. teaches a method wherein polling available redirectors to determine which redirectors are configured to handle the application program's I/O request (see Oehrke et al., col. 8, lines 29-62) and Serlet et al. teaches communicating with at least one other local component to indicate that at least this request can be handled (see Serlet et al., column 5, lines 20-52).

23. As per claim 15, Serlet-Oehrke teach a computer-readable storage medium having computer-executable instructions for performing the method claim 1 (see Serlet et al., column 2, line 51-column 3, line 19).

24. As per claim 16, Serlet et al. teaches a computer-implemented method of automatically and transparently handling WebDAV server and file access requests (see Serlet et al., col. 5, line 60-col. 6, line 14), the method comprising: two or more suitably configured redirectors capable of responding to a WebDAV I/O request (see Serlet et al., col. 6, lines 25-64: wherein the response from the WebDAV/HTTP servers reveals capability status); receiving at a local application programming interface layer an application I/O request comprising a WebDAV Uniform Resource Identifier (URI)

indicating a path and filename of a remote file accessible via WebDAV (see Serlet et al., col. 5, lines 20-52 and col. 7, lines 35-56); and if the specified share and file are accessible, handling the request (see Serlet et al., col. 5, lines 20-52), downloading the file to a local cache of the redirector's file system (see Serlet et al., col. 9, lines 54-63), and returning a file handle corresponding to the file in the local cache to the application program (see Serlet et al., col. 11, lines 24-49). But fails to teach maintaining at a local application programming interface layer a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a WebDAV I/O request, each redirector indicating a configuration suitable for handling the I/O request; polling available redirectors to determine which redirectors are configured to handle the WebDAV URI, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting WebDAV URI requests to corresponding WebDAV server computer systems that store the remote files; receiving responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received WebDAV URI request; determining from the stored priority order which of the plurality of responding redirectors has precedence to handle the WebDAV I/O request; and including, based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the WebDAV I/O request. However, Oehrke et al. teaches maintaining at a local application programming interface layer a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O

request, each redirector indicating a configuration suitable for handling the I/O request (see Oehrke et al., col. 8, lines 29-62); polling available redirectors to determine which redirectors are configured to handle the request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting requests to corresponding server computer systems that store the remote files (see Oehrke et al., col. 8, lines 29-62); receiving responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received request (see Oehrke et al., col. 8, line 63-col. 9, line 2); determining from the stored priority order which of the plurality of responding redirectors has precedence to handle the I/O request (see Oehrke et al., col. 8, lines 29-62) ; and including, based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the I/O request (see Oehrke et al., col. 8, line 63-col. 9, line 2). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. to maintaining at a local application programming interface layer a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O request, each redirector indicating a configuration suitable for handling the I/O request; polling available redirectors to determine which redirectors are configured to handle the request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting requests to corresponding server computer systems that store the remote files; receiving responses from a plurality of suitably configured redirectors,

each suitably configured redirector being capable of redirecting the received request; determining from the stored priority order which of the plurality of responding redirectors has precedence to handle the I/O request; and including, based on the determination, requesting a local file system of the redirector determined to have precedence to create the file in response to the I/O request in order to use redirectors to re-route traffic to other application processors when one processor is unavailable and load balance between available processors (see Oehrke et al., abstract).

25. As per claim 17, Serlet-Oehrke teach the application request includes the Universal Resource Identifier (see Serlet et al., column 5, lines 20-52).

26. As per claim 18, Serlet-Oehrke teach a method wherein the application request includes an identifier that has been previously mapped to at least part of the Universal Resource Identifier (see Serlet et al., column 9, lines 54-63).

27. As per claim 21, Serlet-Oehrke teach a method wherein the application request comprises an I/O request directed to a file, and wherein handling the request comprises creating a local file corresponding to the I/O request (see Serlet et al., column 7, lines 35-56).

28. As per claim 22, Serlet- Oehrke teach downloading at least some file data from the WebDAV server to the local file (see Serlet et al., column 4, lines 27-53: wherein accessing information serves as downloading file data).

29. As per claim 23, Serlet-Oehrke teach returning a file handle corresponding to the local file to the application (see Serlet et al., column 11, lines 24-49).

30. As per claim 26, Serlet-Oehrke teach the application program's request indicates a share on the WebDAV server and further comprising, issuing a WebDAV PROPFIND request directed to the share on the WebDAV server (see Serlet et al., column 11, lines 24-49).

31. As per claim 27, Serlet-Oehrke teach a method wherein the application program's request further indicates a file on the share on the WebDAV server, and further comprising, issuing a WebDAV PROPFIND request directed to the file (see Serlet et al., column 11, lines 24-49).

32. As per claim 32, Serlet-Oehrke teach a computer-readable storage medium having computer-executable instructions for performing the method claim 16 (see Serlet et al., column 2, line 51-column 3, line 19).

33. As per claim 33, Serlet et al. teaches in a computer network, a system for automatically and transparently handling WebDAV server and file access requests (see Serlet et al., col. 5, line 6-col. 7, line 14), the system comprising: an application program that issues WebDAV-related requests, including at least one request having a WebDAV Uniform Resource Identifier (URI) corresponding to path and filename of a remote file stored on a WebDAV server (see Serlet et al., column 5, lines 20-52); or communicating with the WebDAV server to handle requests that cannot be handled locally (see Serlet et al., col. 7, lines 35-56); two or more suitably configured redirectors capable of responding to a WebDAV I/O request (see Serlet et al., col. 6, lines 25-64: wherein the response from the WebDAV/HTTP servers reveals capability status). But fails to teach a WebDAV redirector, the WebDAV redirector configured to respond to polls used to

determine which redirectors are configured to handle the application's WebDAV-related request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting WebDAV file requests to corresponding WebDAV server computer systems that store the remote files; an I/O manager that maintains a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a WebDAV I/O request, each redirector indicating a configuration suitable for handling the I/O request, and that receives responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received WebDAV I/O file request; and determining from the stored priority order which of the plurality of suitably configured redirectors has precedence to handle the WebDAV I/O request and indicating that the WebDAV redirector locally handling each request corresponding to the WebDAV server can be handled locally and was determined to have precedence to create the file in response to the WebDAV I/O request. However, Oehrke et al. teaches a redirector, the redirector configured to respond to polls used to determine which redirectors are configured to handle the application's request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting file requests to corresponding server computer systems that store the remote files; an I/O manager that maintains a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O request, each redirector indicating a configuration suitable for handling the I/O request (see Oehrke et al., col. 8, lines 29-62), and that receives responses from

a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received I/O file request (see Oehrke et al., col. 8, line 63-col. 9, line 2); and determining from a stored priority order which of the plurality of suitably configured redirectors has precedence to handle the I/O request (see Oehrke et al., col. 8, lines 29-62) and indicating that the redirector locally handling each request corresponding to the server can be handled locally and was determined to have precedence to create the file in response to the I/O request (see Oehrke et al., col. 8, line 63-col. 9, line 2). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. to a redirector, the redirector configured to respond to polls used to determine which redirectors are configured to handle the application's request, each redirector suitably configured to handle the I/O request including appropriate functionality for receiving and redirecting file requests to corresponding server computer systems that store the remote files; an I/O manager that maintains a stored a priority order that indicates which of a plurality of redirectors has precedence to handle a I/O request, wherein two or more suitably configured redirectors respond to a I/O request, each redirector indicating a configuration suitable for handling the I/O request, and that receives responses from a plurality of suitably configured redirectors, each suitably configured redirector being capable of redirecting the received I/O file request; and determining from a stored priority order which of the plurality of suitably configured redirectors has precedence to handle the I/O request and indicating that the redirector locally handling each request corresponding to the server can be handled locally and was determined to have precedence to create the file in response to

the I/O request in order to use redirectors to re-route traffic to other application processors when one processor is unavailable and load balance between available processors (see Oehrke et al., abstract).

34. As per claim 34, Serlet-Oehrke teach a system wherein the identifier corresponding to a WebDAV server issued by the application comprises a Universal Resource Identifier (see Serlet et al., column 5, lines 20-52).

35. As per claim 35, Serlet-Oehrke teach a system wherein the identifier corresponding to a WebDAV server issued by the application comprises an identifier previously mapped to a share on the WebDAV server (see Serlet et al., column 9, lines 54-63).

36. As per claim 38, Serlet-Oehrke teach a system that: creates a local representation of the file (see Serlet et al., column 6, line 65-column 7, line 34); determines whether the file exists on the WebDAV server, and if so, downloads at least some of the data from the WebDAV server file to the local representation of the file (see Serlet et al., column 4, lines 27-53: wherein accessing information serves as downloading file data); returns a file handle corresponding to the local representation of the file to the application program (see Serlet et al., column 11, lines 24-49); receives I/O read and write requests associated with the file handle and handles the I/O read and write requests via the local representation of the file (see Serlet et al., column 11, lines 24-49; column 12, lines 35-44); and receives an I/O close request associated with the file handle, and handles the I/O close request by closing the local representation of the

file and uploading at least part of the local representation of the file to the WebDAV server (see Serlet et al., column 11, lines 24-49).

37. Claims 12-14, 28-30, and 39-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Serlet et al. and Oehrke et al. as applied to claims 1 and 16 above, and further in view of Prust (6,714,968).

38. As per claim 12, Serlet et al. and Oehrke et al. teach the mentioned limitations of claim 1 above, but fail to teach determining that the file is encrypted on the WebDAV server, and wherein downloading the file to a local cache comprises, communicating with the file system to create an image of the file in the local cache that is also encrypted. Prust teaches determining that the file is encrypted on the WebDAV server, and wherein downloading the file to a local cache comprises, communicating with the file system to create an image of the file in the local cache that is also encrypted (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add determining that the file is encrypted on the WebDAV server, and wherein downloading the file to a local cache comprises, communicating with the file system to create an image of the file in the local cache that is also encrypted in order to allocate a corresponding storage area for each user and store the respective user information in metadata database (see Prust, col. 7, line 59-col. 8, line 7).

39. As per claim 13, Serlet et al., Prust, and Oehrke et al. teach the mentioned limitations of claims 1 and 12 above, but Serlet et al. and Oehrke et al. fail to teach communicating with the file system to open the image of the file such that the file system will transparently decrypt file data on read requests and will transparently encrypt file data on write requests to the file. Prust teaches communicating with the file system to open the image of the file such that the file system will transparently decrypt file data on read requests and will transparently encrypt file data on write requests to the file (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add communicating with the file system to open the image of the file such that the file system will transparently decrypt file data on read requests and will transparently encrypt file data on write requests to the file in order to allow the user to access the respective storage area via the many access interfaces (see Prust, col. 7, line 59-col. 8, line 7).

40. As per claim 14, Serlet et al., Prust, and Oehrke et al. teach the mentioned limitations of claims 1 and 12 above, but Serlet et al. and Oehrke et al. fail to teach uploading the file from the local cache to the WebDAV server comprises, communicating with the file system to read data from the local image of the file such that the file will be uploaded as the encrypted image thereof. Prust teaches uploading the file from the local cache to the WebDAV server comprises, communicating with the file system to read data from the local image of the file such that the file will be uploaded as the encrypted image thereof (see Prust, column 7, lines 39-55: wherein encryption

and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add uploading the file from the local cache to the WebDAV server comprises, communicating with the file system to read data from the local image of the file such that the file will be uploaded as the encrypted image thereof in order to prevent unauthorized users from accessing information about other users (see Prust, column 7, lines 39-55).

41. As per claim 28, Serlet et al. and Oehrke et al. teach the mentioned limitations of claim 16 above but fail to teach the application request comprises an I/O request directed to an encrypted file, and further comprising, automatically decrypting the data locally when downloading the encrypted file from the WebDAV server and automatically encrypting the data locally when uploading the encrypted file to the WebDAV server. However, Prust teaches the application request comprises an I/O request directed to an encrypted file, and further comprising, automatically decrypting the data locally when downloading the encrypted file from the WebDAV server and automatically encrypting the data locally when uploading the encrypted file to the WebDAV server (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add the application request comprises an I/O request directed to an encrypted file, and further comprising, automatically decrypting the data locally when downloading the encrypted file from the WebDAV server and automatically encrypting the data locally when uploading the encrypted file to the WebDAV server in

order to allocate a corresponding storage area for each user and store the respective user information in metadata database (see Prust, col. 7, line 59-col. 8, line 7).

42. As per claim 29, Serlet et al. and Oehrke et al. teach the mentioned limitations of claim 16 above, but fail to teach the application request comprises an I/O request directed to a file that is encrypted on the WebDAV server, and wherein handling the request comprises, creating a local file corresponding to the I/O request and downloading an image of the file on the WebDAV server to the local file, wherein the local file is written by a local system such that the image corresponds to the encrypted image on the WebDAV server. However, Prust teaches the application request comprises an I/O request directed to a file that is encrypted on the WebDAV server, and wherein handling the request comprises, creating a local file corresponding to the I/O request and downloading an image of the file on the WebDAV server to the local file, wherein the local file is written by a local system such that the image corresponds to the encrypted image on the WebDAV server (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add the application request comprises an I/O request directed to a file that is encrypted on the WebDAV server, and wherein handling the request comprises, creating a local file corresponding to the I/O request and downloading an image of the file on the WebDAV server to the local file, wherein the local file is written by a local system such that the image corresponds to the encrypted image on the WebDAV server in order to

allow the user to access the respective storage area via the many access interfaces (see Prust, col. 7, line 59-col. 8, line 7).

43. As per claim 30, Serlet et al., Prust, and Oehrke et al. teach the mentioned limitations of claims 16 and 29 above, but Serlet et al. and Oehrke et al. fail to teach communicating with the file system to open the local file such that the file system will transparently decrypt file data read on read requests and will transparently encrypt file data written on write requests. Prust teaches communicating with the file system to open the local file such that the file system will transparently decrypt file data read on read requests and will transparently encrypt file data written on write requests (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written). It would have been obvious to one having ordinary skill in the art at the time of the invention to add communicating with the file system to open the local file such that the file system will transparently decrypt file data read on read requests and will transparently encrypt file data written on write requests in order to store data files and communicate the data files to the storage server for storage within the storage area (see Prust, col.1, lines 49-67).

44. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Serlet et al., Oehrke et al., and Prust (6,714,968). Serlet et al., Oehrke et al., and Prust teach the limitations mentioned above in claims 16, 29, and 30 but Prust and Oehrke et al. fail to teach detecting a request to close the local file, closing the local file, communicating with the file system to open the local file such that the file will not be decrypted when read and uploading the file to the WebDAV server as an encrypted file. However, Serlet

et al. teaches detecting a request to close the local file, closing the local file, communicating with the file system to open the local file (see Serlet et al., column 11, lines 24-49); such that the file will not be decrypted when read (see Serlet et al., column 12, lines 35-44); and uploading the file to the WebDAV server as an encrypted file (see Serlet et al., column 5, line 60-column 6, line 14: wherein authenticated access functions as being encrypted). It would have been obvious to one having ordinary skill in the art at the time of the invention to add detecting a request to close the local file, closing the local file, communicating with the file system to open the local file such that the file will not be decrypted when read and uploading the file to the WebDAV server as an encrypted file in order to allow only the authorized user to have access to his/her data on the WebDAV server (see Serlet et al., col. 6, lines 15-23).

45. As per claim 39, Serlet et al. and Oehrke et al. teach the limitations mentioned above in claims 33 and 38 and furthermore teaches a system wherein requesting the file system to create a local file that is opened such that transparent encryption and decryption are not enabled therefor (see Serlet et al., column 5, line 60-column 6, line 14: wherein authenticated access may not be enabled by the user); requesting the file system to close the local file (see Serlet et al., column 11, lines 24-49). But fails to teach the WebDAV file is encrypted, and wherein downloading at least some of the encrypted file data by requesting the file system to write to the local file without translation thereof. Prust however teaches the WebDAV file is encrypted (see Prust, column 7, lines 39-55: wherein encryption and decryption may be done either when the file is read or written); downloading at least some of the encrypted file data by requesting the file system to

write to the local file without translation thereof (see Prust, column 7, lines 7-34). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. to a system wherein the WebDAV file is encrypted, and that creates the local representation of the file by downloading at least some of the encrypted file data by requesting the file system to write to the local file without translation thereof in order to allow an user to access virtual storage area using a conventional electronic mail software application (see Prust, column 7, lines 7-34).

46. As per claim 40, Serlet et al., Oehrke et al., and Prust teach the limitations mentioned above in claims 33, 38, and 39. Serlet et al. also teaches a system requesting the file system to reopen the local file (column 11, lines 24-49). But fails to teach reads therefrom are decrypted and writes thereto are encrypted. Prust however teaches reads therefrom are decrypted and writes thereto are encrypted (column 7, lines 39-55). It would have been obvious to one having ordinary skill in the art at the time of the invention to add reads therefrom are decrypted and writes thereto are encrypted in order to allocate a corresponding storage area for each user and store the respective user information in metadata database (see Prust, col. 7, line 59-col.8, line 7).

47. As per claim 41, Serlet et al., Oehrke et al., and Prust teach the limitations mentioned above in claims 33, 38, 39, and 40. But Oehrke et al. and Prust fail to teach when the WebDAV redirector handles the I/O close request, and before uploading the file, the WebDAV redirector closes the local representation of the file, and reopens the local file by requesting the file system to open the file such that reads therefrom are not

decrypted. Serlet et al. however teaches a method of handling the I/O close request, and before uploading the file, closing the local representation of the file (see Serlet et al., column 11, lines 24-49), and reopening the local file by requesting the file system to open the file such that reads therefrom are not decrypted (see Serlet et al., column 5, line 60-column 6, line 14: wherein authenticated access function as being encrypted). It would have been obvious to one having ordinary skill in the art at the time of the invention to add a method of handling the I/O close request, and before uploading the file, closing the local representation of the file, and reopening the local file by requesting the file system to open the file such that reads therefrom are not decrypted in order for authorized users to access their data on the WebDAV server without needing to input authentication information for every transmission (see Serlet et al., col. 6, lines 15-23).

48. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Serlet et al. and Oehrke et al. as applied to claim 16 above, and further in view of Charisius et al. (2002/0078432). Serlet et al. and Oehrke et al. teach the mentioned limitations of claim 16 above, but fail to teach a networking request to browse a network share on the WebDAV server, and wherein handling the request includes enumerating information of the network share. However, Charisius et al. teaches a networking request to browse a network share on the WebDAV server, and wherein handling the request includes enumerating information of the network share (see Charisius et al., ¶132). It would have been obvious to one having ordinary skill in the art at the time of the invention to add a networking request to browse a network share on the WebDAV server, and wherein

handling the request includes enumerating information of the network share in order to allow more than one user to view the same workflow or project plan, to provide persistent storage, to monitor the progress of an activated project plan, to simultaneously create plans from the same workflow, and to have essentially unlimited access to the power of the tool through the ubiquity of the Internet (see Charisius et al., ¶ 10).

49. Claims 36 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Serlet et al. and Oehrke et al. as applied to claim 33 above, and further in view of French (6,654,794).

50. As per claim 36, Serlet et al. and Oehrke et al. teach the above-mentioned limitations of claim 33, but fail to teach a system wherein the WebDAV redirector receives requests from the application via an application programming interface. However, French teaches a system wherein the WebDAV redirector receives requests from the application via an application programming interface (see French, col. 4, line 66-col. 5, line 19). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. and Oehrke et al. to a system wherein the WebDAV redirector receives requests from the application via an application programming interface in order to perform its various operations and to provide the requisite functionality of its features (see French, col. 4, line 66-col. 5, line 19).

51. As per claim 37, Serlet et al. and Oehrke et al. teach the above-mentioned limitations of claim 33, but fail to teach a system wherein the WebDAV redirector

receives the I/O request from a manager component. However, French teaches a system wherein the WebDAV redirector receives the I/O request from a manager component (see French, col. 4, lines 58-65). It would have been obvious to one having ordinary skill in the art at the time of the invention to modify Serlet et al. and Oehrke et al. to a system wherein the WebDAV redirector receives the I/O request from a manager component in order to order to perform its various operations and to provide the requisite functionality of its features (see French, col. 4, line 66-col. 5, line 19).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ranodhi Serrao whose telephone number is (571)272-7967. The examiner can normally be reached on 8:00-4:30pm, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on (571)272-3880. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

RNS

R.N.S.

1/31/2008

WILLIAM VAUGHN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100